Decidable fragments of first order logic

R. Ramanujam

The Institute of Mathematical Sciences, Chennai, India

jam@imsc.res.in
Modal logics have decent algorithmic properties, useful for specification and verification.

Vardi, 1996: Why are modal logics so robustly decidable?

Perhaps because they sit inside the two-variable fragment of First order logic?

Andreka, van Benthem, Nemeti: Because they correspond to a guarded fragment of First order logic.

Some strong evidence, thanks to the work of Erich Grädel, Martin Otto and some co-authors.
The decision problem

- **David Hilbert**: Find an algorithm which, given any first order sentence, determines whether it is satisfiable.
- **Bernays, Schönfinkel, 1928**: $\exists^* \forall^*$, without equality, but no function symbols.
- **Ramsey 1928**: class above, with equality.
- **Ackermann 1928**: $\exists^* \forall^* \exists^*$.
- **Gödel, Kalmár, Schutte 1932-34**: $\exists^* \forall^2 \exists^*$, without equality.
Undecidability

- Church, Turing 1936: The satisfiability problem for first order logic is algorithmically unsolvable.
- Trakhtenbrot 1950: Satisfiability over finite structures is undecidable.
- Hence the class of formulas valid over finite structures is not recursively axiomatizable.
- Shift, from decision problem, to classification problem.
Prefix classes

- **Kalmár, Suranyi 1950’s**: With one binary relation, and without equality, $\forall^* \exists$ is undecidable, as also: $\exists^* \forall^3 \exists^*$, $\exists^* \forall \exists \forall$.

- **Gurevich 1976**: With no relational symbols, but with two function symbols and equality, the class $\forall$ is undecidable.

- **Goldfarb 1984**: The Gödel class is undecidable in the presence of equality.

- **Goldfarb, Gurevich, Rabin, Shelah**: all decidable and undecidable prefix classes completely characterized.
Why prefix classes?

- Historical: early results were for prefix classes.
- Natural syntactic fragments; helped focus on role of equality.
- Classification of mathematical theories, especially those of groups, rings and fields.
- Modern understanding of blocks of quantifiers in descriptive complexity.
Modal logic

Simplest logic: $\langle a \rangle \alpha, [a]\alpha$, $a \in \Sigma$, a finite set. Has good model theoretic and algorithmic properties.

- Fragment of first order logic.
- Map $\alpha$ to $\alpha^*$ of FOL:
  
  $\langle a \rangle \alpha \rightarrow \exists y : (E_a(x, y) \land \alpha^*(y))$
  
  $[a]\alpha \rightarrow \forall y : (E_a(x, y) \implies \alpha^*(y))$

- Satisfiability: PSpace-complete.
- Model checking: $O(\mathcal{K} \cdot \alpha)$. 

Update meeting TRDDC, July 17-19, 2008
Limitations of modal logic

Modal logic is very weak in terms of expressive power.

- **No equality**: We cannot say that both an $a$-transition and $b$-transition from the current state lead us to the same state.

- **Bounded quantification**: We cannot say that a property holds in all states.

- **New transitions not definable**: For instance, we cannot define $E(x, y) = E_a(y, x) \land E_b(y, x)$. 
More limitations

More on the list of complaints.

- **No counting:** We cannot say that there is at most one $a$-transition from the current state (and hence cannot distinguish deterministic systems from nondeterministic ones.

- **No recursion:** We can look only at a bounded number of transition steps. This is a limitation shared by FOL as well.

And yet, modal logic is interesting, on many counts.
In praise of modal logic

It has interesting model theoretic properties.

► Invariance under bisimulation:

\[(\mathcal{K}, w \models \alpha \land (\mathcal{K}, w) \sim (\mathcal{K}', w') \implies (\mathcal{K}', w') \models \alpha)\]

► In fact, ML is the bisimulation invariant fragment of FOL.

► It has the finite model property.

► It has the tree model property.
Extensions

Numerous extensions of ML, designed to overcome the limitations mentioned, still with similar model theoretic and algorithmic properties.

- $PDL = ML +$ transitive closure.
- $LTL = ML +$ temporal operators on paths.
- $CTL = ML +$ temporal operators on paths + path quantification.
- $\mu$-calculus: encompasses these and others like game logics and description logics.
Robustness

All these extensions have good algorithmic properties. The following hold for the $\mu$-calculus, which encompasses most modal logics of computation.

- Satisfiability is Exptime-complete.
- Efficient model checking for many subclasses; in general, is in $NP \cap co-NP$.
- Bisimulation invariant fragment of monadic second order logic.
Vardi’s question

- Vardi, 1996: Why are modal logics so robustly decidable?
- The standard translation from ML to FO does not need more than two free variables.
- Traditionally, this has been used as an explanation for why ML has good properties.
- Is this explanation convincing?
Fixed variable FO

$\mathit{FO}^k$: relational fragment of FOL with only $k$ free variables.

- "There exists a path of length 17" is in $\mathit{FO}^2$:

\[ \exists x \exists y (E(x, y) \land \exists x (E(x, y) \land \exists y (E(x, y) \land \ldots \exists y E(x, y)) \ldots )) \]

- The satisfiability problem is undecidable for $\mathit{FO}^k$, for all $k \geq 3$.

- This is true even for most of the prefix classes.
Two variable FO

- **Scott 1962**: $\mathit{FO}^2$ without equality can be reduced to the Gödel class and is hence decidable.

- **Mortimer 1975**: $\mathit{FO}^2$ has the finite model property, and is decidable.

- In fact, if $\phi \in \mathit{FO}^2$ is satisfiable, then it is satisfiable in a model whose size is at most doubly exponential in the size of $\phi$.

- **Grädel, Kolaitis, Vardi, 1997**: $\mathit{FO}^2$ satisfiability is $\mathsf{NExpTime}$ complete. (Lower bound essentially from Fürer 1981.)
Not robust

\(FO^2\) is not nearly as robustly decidable as modal logic.

- Grädel, Otto, Rosen, 1999: \(FO^2 +\) transitive closure is undecidable, as also \(FO^2 +\) path quantification, or \(FO^2 +\) fixed point operators.
- In fact, they are (typically) \(\Sigma_1^1\)-hard.
What ails $FO^2$?

- Modal logics typically have the tree model property: every satisfiable formula has a model that is a tree.
- In fact, the tree is boundedly branching.
- $FO^2$ lacks this property: consider the sentence $\forall x \forall y. E(x, y)$.
- Most of the extensions mentioned can encode grids.
Why trees?

Finite model property many mean decidability, but why bother to have a tree model property?

- Typically tree models allow the use of powerful tools. For $\mu$-calculus, we can interpret them in the monadic second order theory of the infinite tree and use Rabin’s theorem.
- This reduction gives decidability but not good complexity.
- However, the proof of Rabin’s theorem uses tree automata, and by constructing tree automata directly, we get good algorithms.
- $FO^2$ is not the answer to Vardi’s question.
A closer look

A closer look at the translation from ML to FOL shows not only the use of two variable logic, but also $\exists x. (E_a(x, y) \land \ldots)$ and $\forall x. (E_a(x, y) \implies \ldots)$.

- Thus quantifiers are always relativized by atoms in the modal fragment of FOL.
- Each subformula can ”speak” only about elements that are ‘close together’ or guarded.
- **Guarded fragment**: Quantification is of the form: $\exists x. (\alpha(x, y) \land \phi(x, y))$ and $\forall x. (\alpha(x, y) \implies \phi(x, y))$. $\alpha$ is atomic and contains all the free variables in $\phi$. 
A challenge

- Andréka, van Benthem, Nemeti 1998: The guarded nature of quantification in modal logics is the "real" reason for their good algorithmic and model theoretic properties.
- Results proved since then provide some positive evidence.
The definition

\[ GF, \text{ the guarded fragment of FOL is the least set of formulas such that:} \]

1. Every relational \( R(x_1, \ldots, x_m) \) and \( x = y \) are in \( GF \).
2. \( GF \) is closed under boolean connectives.
3. If \( x, y \) are tuples of variables, \( \alpha(x, y) \) is a positive atomic formula, and \( \phi(x, y) \) is in \( GF \) such that \( \text{free}(\phi) \subseteq \text{free}(\alpha) \subseteq (x \cup y) \), then the formulae \( \exists x.(\alpha(x, y) \land \phi(x, y)) \) and \( \forall x.(\alpha(x, y) \implies \phi(x, y)) \) are also in \( GF \).
Extension of ML

It is clear that ML maps into GF, but do we have more?

- There are no restrictions on using monadic or binary predicates.
- We have equality.
- We can define new transition relations.
- No strict separation between state properties and transitions.
Good news on GF

- Decidable (Andréka, van Benthem, Németi).
- Has the finite model property (Andréka, Hodkinson, Németi).
- Has a tree model (like) property: every satisfiable formula has a model of small tree width (Grädel).
- Satisfiability is 2-Exptime complete, and for formulas of bounded arity, Exptime complete (Grädel).
- Has efficient game based model checking algorithms.
- GF is invariant under guarded bisimulation (van Benthem).
Need for extensions

Examples of FO properties not in GF.

- Transitivity, as also "Between-ness": all points between $x$ and $y$ have property $\phi(y)$.
- Note that the latter property is typically needed for temporal logics.
- Guards in both behave differently; "Between-ness" needs conjunctions of atoms.
- Loosely guarded fragment: conjunctive guards. LGF has most of the nice properties and is decidable.
- More decidable extensions recently (clique-guarded, action-guarded etc).
- But GC + transitive closure is undecidable.
Guarded fixed point logic

$GF$ is robustly decidable.

- Grädel, Walukiewicz 1999: $\mu \models GF$, an extension of $GF$ with fixed-point operators is decidable.
- $\mu \models GF$ does not have finite model property, but has models that have small tree width.
- Complexity is the same as for $GF$. 

Update meeting TRDDC, July 17-19, 2008
Definition of $\mu - GF$

Let $R$ be a $k$-ary relation variable, and $x$, a $k$-tuple of distinct variables.
Let $\phi(R, x)$ be a guarded formula where $R$ appears only positively and not in guards and contains no free variables outside $x$.
Then $[\mu R x . \phi](x)$ and $[\mu R x . \phi](x)$ are in $\mu - GF$. 

An example

\( \mu - GF \) formulas are not easy to parse!

- \( \exists xy. F(x, y). \)
- \( \forall xy. (F(x, y) \implies \exists x. F(y, x)). \)
- \( \forall xy. (F(x, y) \implies [\mu Rx. \forall y (F(y, x) \implies Ry)](x)). \)

In the last formula, the lfp is the set of points that have only finitely many predecessors. Thus, the sentence says that there is an infinite forward \( F \)-chain, but no backward \( F \)-chain. Specifically, there is no \( F \)-cycle.
The tree property

\( \mu_G F \) models are (of course) not trees, but structures of small tree width.

A structure has tree width \( k \) if it can be covered by a tree-shaped arrangement of substructures of size at most \( k + 1 \).

The tree width of a structure measures how closely it resembles a tree.

- Forests have tree width 1.
- Cycles have tree width 2.
- Finite rectangular grids have unbounded tree width.
A proof technique

We can use Rabin’s theorem to get decidability but need tree automata to get decent complexity (alternating two-way tree automata with parity acceptance condition).

- But generally we need boundedly branching trees to apply tree automata.
- Etessami, Wilke 2005: Technique to use alternating automata on arbitrary branching trees.
- Automaton treats all edges at current node (as also the edge to parent) in the same way.
- A general forgetful determinacy theorem for games on graphs used to show that it automaton accepts a tree then it also accepts one that is boundedly branching.
Guarded logics

- Liberal guardedness conditions leading to more expressiveness.
- Guarded fragments of other logics (like "Datalog-Lite"), and second order logics.
- Decidable fragments on structures where two variable logic is undecidable.
- Applicable to arbitrary relational structures.
- Hope for decidable logics on partial orders.